View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

Mitigating the Effects of Partial Resource Failures for Cloud Providers

Tim Püschel
Albert-Ludwigs-Universität Freiburg, tim.pueschel@is.uni-freiburg.de

Dirk Neumann
Albert-Ludwigs-Universität Freiburg, dirk.neumann@is.uni-freiburg.de

Follow this and additional works at: http://aisel.aisnet.org/amcis2010

Recommended Citation
Püschel, Tim and Neumann, Dirk, "Mitigating the Effects of Partial Resource Failures for Cloud Providers" (2010). *AMCIS 2010 Proceedings*. 580.
http://aisel.aisnet.org/amcis2010/580

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Mitigating the Effects of Partial Resource Failures for Cloud Providers

Tim Püschel
Chair for Information Systems Research,
Albert-Ludwigs-Universität Freiburg,
Germany
tim.pueschel@is.uni-freiburg.de

Dirk Neumann
Chair for Information Systems Research,
Albert-Ludwigs-Universität Freiburg,
Germany
dirk.neumann@is.uni-freiburg.de

ABSTRACT

Competition for users on a global market is fierce, forcing enterprises to provide for better, faster services while offering the same more cheaply. At the same time, users choose to remain oblivious of the infrastructure behind the service – only demanding that it works. Cloud service failures and inefficient management of such failures can result in significant financial cost, loss of reputation for providers, and drive key customers away. At the same time failure situations can never be completely avoided. To mitigate their effects we present a decision model for providers to help them decide which jobs to keep running and which to cancel in order to minimize loss of revenue and key customers during partial resource failures. The results of the evaluation of the model and its extension show its ability to significantly improve revenue. Furthermore the model can also help to reduce the number of cancelled jobs.

Keywords

Cloud Computing, Failure Management,

INTRODUCTION

As technology simplifies our way of life at an increasing rate, we often begin to take things for granted and no longer care about the how and the where. Where accessing our e-mail account over a web browser anywhere in the world began as a novelty, nowadays we expect nothing less from other services, even worse – we want faster and better services. As a result, the industry provides new and better services ranging online office applications or video delivery to infrastructure services such as online storage or virtual machines.

Competition on a global market forces enterprises to provide for better, faster services while offering the same more cheaply. To achieve this goal it is necessary to maintain a very high degree of flexibility with respect to the IT infrastructure (Carr 2005).

Facing this challenge the ideas of Cloud Computing and "Infrastructure-/Platform-/Software-as-a-Service" became of interest. The term Cloud Computing, initially coined by Amazon.com which were among the first companies to offer Cloud services on a large scale, depicts the offering of resources (e.g. processing power, storage and bandwidth) bundled as a services, which are offered to users often oblivious of the infrastructure beneath.

In short, Cloud Computing allows the introduction of new products and services without requiring large investments in upgrades, or installation of new IT equipment. As an example, the New York Times managed to convert 4TB of scanned images containing 11 million articles into PDF files in very little time with Hadoop, a framework for distributed applications using Amazon's Cloud services. The actual conversion process took only 24 hours costing merely 240 US$ for processing (New York Times Blog 2007). The makers of SmugMug, a photo sharing website, estimate their savings using Amazons Simple Storage Service at about 500,000 US$ per year (Business Week 2006). According to analysts at Gartner "The projected shift to Cloud Computing will result in dramatic growth in IT products in some areas and in significant reductions in other areas." (Gartner 2008).

The more providers offer their resources or services, the more likely it is that customers can access them at competitive prices and quality. In this regard, it is important to attract more providers. At the same time basic infrastructure services, such as content delivery or provisioning of virtual machines, offered by different provides are relatively similar. This allows for

increasing competition, which often results in lower profit margins for providers. Technologic advancements are well distributed in the hardware, meaning all cloud service providers often have access to similar equipment. This means that profit margins can only be increased by improving efficiency of Cloud infrastructure management.

However, for customers planning to replace some of their infrastructure with cloud services price is usually not the only criteria. Availability and reliability of the services, as well as the manner in which cloud provider deals with outages (e.g. by paying penalties) also play a vital role. Due to the novelty of the market, providers have not yet established risk management systems. However as users and businesses rely more and more on cloud services the management of resource failures and mitigation of effects of outages can become a key advantage in prospect of fierce competition. Overload situations can lead to reduced overall performance (Nou et al. 2007) and thereby can result in breaking service level agreements between the provider and clients.

To improve performance in the commercialization of distributed computational resources decisions about the supplied resources and their management should be based on both technical and economic aspects (Kenyon and Cheliotis 2004). This applies not only to resource or Service-Level-Agreement (SLA) management under normal conditions but also holds true when problems arise. With state-of-the-art technology, this assimilation is hampered, as the resource managers facilitating the deployment of the resources are not designed to incorporate economic issues (e.g. penalties for job cancellation).

Technical resource management systems typically offer the possibility to include priorities for user groups. In purely revenue maximizing mechanisms it is not possible to distinguish important from unimportant partners, as only current prices matter for the allocation. However for many companies this can play an important role when deciding which already ongoing jobs to cancel in overload situations to maintain system stability.

This leads to the overall research question: Which jobs or service requests should Cloud providers cancel when faced with partial resource failure in order to comply with their business policies and maximize revenue? To solve this issue providers need a decision model helping them to make this decision.

In this work it will be motivated that that economic aspects need to be taken into account when taking this decision. We will introduce a basic revenue maximizing decision model for providers as well as two extensions dealing with rental of additional capacity elsewhere as well as client classification. We will show how these models can be used to improve the decisions taken in overload situations.

Essentially, there are three main reasons for the integration of client classification: First, it allows giving internal users the necessary priorities to maintain their service levels. Second, it permits the inclusion of long-term oriented relationships with strategically important customers so-called credential components. Finally, client classification can be used as an instrument of revenue management, which allows skimming off consumer surplus. The integration of dynamic pricing allows giving customers incentives to run their jobs during times of low utilization and thereby achieve a more even utilization. Furthermore it can help to achieve higher prices and therefore higher revenue in times of high demand.

The example of Amazon.com as a Cloud provider can be used to illustrate some of the challenges for Cloud providers. Amazon maintains a Cloud infrastructure that is used by its own shopping website but also offers different services, such as S3 or EC2 for external users. This allows balancing usage spikes to some extent and realizing economies of scale in the maintenance of its infrastructure. However, the sale of services to external users should not have negative impacts on its core business. Especially it should not result in decreased resource availability for its shopping site. These effects can be avoided by maintaining a large buffer of spare resources or by an effective capacity and resource management.

RELATED WORK

There are several research streams dealing with client classification, Quality of Service and risk management for service providers. The first one comes from the research area of computer science and focuses on the technical aspects while incorporating some economic aspects. Another stream aims to adapt concepts and from Revenue Management to job admission problem. Most existing work applies these concepts to Grid computing, Utility computing or the Software-as-a-Service business.

Elements of client classification such as price discrimination based on customer characteristics have been mentioned in other papers (Newhouse et al. 2004 and Buyya 2002). They did however not consider other discrimination factors. Chicco et al. (2006) describe data-mining algorithms and tools for client classification in the electricity grids but concentrate on methods for finding groups of customers with similar behavior. An architecture for admission control on e-commerce websites that prioritizes user sessions based their intentions to buy a product is proposed by Poggi et al. (2007 and 2009). They analyzed customer behavior, such as navigational clicks, on an e-commerce web site. Based on this behavior predictions about the

user's intention to buy a product can be made. Consequently, Quality of Service for user sessions is shaped based the user's intentions.

Boughton et al. (2006) present research on how workload class importance can be considered for low-level resource allocation. They focus on competing workloads in databases and investigate who business policies describing the relative importance of workloads can be used to efficiently allocate resources.

One approach to realize end-to-end Quality of Service is the Globus Architecture for Reservation and Allocation (Foster et al. 1999). This approach uses advance reservations to achieve QoS. Another way to achieve autonomic QoS aware resource management is based on online performance models (Kounev et al. 2007). They introduce a framework for designing resource managers that are able to predict the impact of a job in the performance and adapt the resource allocation in such a way that SLAs can be fulfilled. Both approaches do not consider achieving QoS in case of partial resource failure.

The introduction of risk management to the Grid (Djemame et al. 2006) permits a more dynamic approach to the usage of SLAs. It allows modeling the risk that the SLA cannot be fulfilled within the service level agreement. A provider can then offer SLAs with different risk profiles. However, such risk modeling can be very complex. It requires information about the causes of the failure and its respective probabilities. Clients need to have the possibility to validate the accuracy and correctness of the providers risk assessment and risks have to be modeled in the SLAs. Voss (2006) proposes the use of precautionary migrations of tasks for preventing SLA violations, while emphasizing risk management strategies.

SERVICE/JOB CANCELLATION MODEL

The key question in light of partial resource failure or other overload situations is which currently running services or jobs to cancel in order to maintain system stability and fulfill part of the SLAs. If this question is not addressed and no jobs are cancelled very often none of the services has enough resources available to maintain its SLA. One simple approach is to randomly cancel jobs until enough capacity is available. Obviously this usually still does not result in adequate solutions. In this section we present a decision model to address this question. Our basic model focuses on economic aspects such as the prices customers are paying and penalties due for cancellation of jobs. We then extend our model to take the possibility of renting additional capacity and migrating some of the jobs there into account. The second extension of our model deals with client classification.

Basic Model: Revenue Maximization

Assuming the provider is only interested to maximize its revenue, has no possibilities to acquire additional capacity, and there are no further constraints, a problem can be described as follows:

$$\max_{x} \sum_{j \in J} x_j * fp_j - (1 - x_j) * s_j \qquad \text{(O1)}$$

Subject to:

$$\sum_{j \in J} c_{jr}(t) * x_j \leq c_r(t) \forall t \in T, \forall r \in R \qquad \text{(C1)}$$

Where T is the set of all regarded timeslots; J is the set of jobs currently running; R is the set of all resource types; fp_j is the price paid for job j; s_j is the penalty which has to be paid for cancellation of job j; x_j is a binary allocation variable indicating whether job j was accepted or rejected; $c_{jr}(t)$ is the capacity required by job j in timeslot t; and $c_r(t)$ is the total capacity available for resource type r during timeslot t.

(O1) is the objective function and represents the achieved revenue. (C1) represents the capacity constraint for the different types of resources (e.g. CPU, memory, storage, bandwidth, …). It assures that not more capacity can be allocated than is available. Running jobs or service requests can have different resource usage profiles, such as cpu intensive or memory jobs.

The solution for this problem can be calculated with standard solvers. After the solution for this problem is calculated all jobs with $x_j=0$ are cancelled. This model essentially applies the idea of dynamic pricing to a cancellation scenario. Those jobs that pay higher prices or have high cancellation penalties are kept, while jobs delivering lower revenue are cancelled.

First Extension: Renting Additional Resources

If the provider can rent additional capacity for a certain fee, the maximization problem can be adapted as follows:

$$\max_x \sum_{j \in J} x_j * (fp_j - x_{jc} * of_j) - (1 - x_j) * s_j \qquad \text{(O1.1)}$$

Subject to:

$$\sum_{j \in J} c_{jr}(t) * x_j * (1 - x_{jc}) \leq c_r(t) \forall t \in T, \forall r \in R \qquad \text{(C1.1)}$$

$$\sum_{j \in J} c_{jr}(t) * x_j \leq c_r(t) + oc_r(t) \forall t \in T, \forall r \in R \qquad \text{(C2.1)}$$

Where of_j is the fee the provider has to pay for renting the capacity necessary for job j; x_{jc} is a binary allocation variable which is 1 for all jobs which will run on rented capacity and 0 for all other jobs; $oc_r(t)$ is the capacity which can be rented for resource type r during timeslot t.

The objective function (O1.1) is slightly modified from the basic model to include the cost for renting the additional capacity. Constraint (C1.1) states that the resources required for jobs allocated to own resources have to be smaller or equal to the own capacity available. (C2.1) states that resources required for all allocated jobs have to be smaller or equal to the total capacity available, i.e. own and rented resources.

After the solution for this problem is calculated all jobs with x_j=0 are cancelled; all jobs with x_{jc}=1 are migrated to rented resources. In real world situations this could be hindered by compatibility issues. While the growing success and deployment of virtualization technologies mitigate these issues it might still be necessary to build software compatibility layers.

Depending on the rental fees and the available capacity which can be rented this extension can result in no jobs being cancelled and all moved to rented capacity. However it is also possible that only few or no jobs are moved to rented resources.

Second Extension: Client Classification

In both of the above cases we assumed that the provider is only interested in short term revenue maximization. However in many situations providers can have additional constraints. In some cases for example it might be necessary to give some of the users or customers certain privileges.

The following model assures that certain or all jobs from important customers, from now on called "gold users", receive priority. This means that first all of these jobs need to be preserved and if there is capacity left afterwards other jobs are considered. This can be achieved with the following model:

$$\max_x \frac{\sum_{j \in J} cc_j * (x_j * fp_j - (1 - x_j) * s_j)}{\sum_{j \in J} cc_j * fp_j} * m + \frac{\sum_{j \in J} (1 - cc_j) * (x_j * fp_j - (1 - x_j) * s_j)}{\sum_{j \in J} (1 - cc_j) * fp_j} \qquad \text{(O2)}$$

Subject to:

$$\sum_{j \in J} c_{jr}(t) * x_j \leq c_r(t) \forall t \in T, \forall r \in R \qquad \text{(C1)}$$

Where cc_j is a binary variable indicating whether job j is from a gold user or not; m determines whether gold users should always be preferred or whether there should be a tradeoff between priority and revenue maximization. A value of 1 for m

represents no priority; values larger than one represent increased priority; values smaller than result in undervaluation of jobs with the cc_j flag.

Objective function (O2) includes client classification but does not consider the possibility to rent additional capacity introduced in (O1.1).

It is expected that this extension results in slightly lower revenue than the basic model and first extension. However, the ratio of gold jobs cancelled will be significantly lower.

EVALUTATION

For the evaluation first ten different scenarios where stochastically generated. In each scenario the cloud provider has about 15 jobs running concurrently. Each job has different resource requirements, prices and penalties, due in case of cancellation. Workload models (Feitelson 2009) can be used for generation of resource requirements. Regular service prices or spot prices (eg. Amazon EC2 spot prices) can be used as a basis for generation of prices. For the generation of values for the penalties however no such extensive information is available.

The three types of resources considered in this setting are CPU, memory and bandwidth. It was then assumed that at some point part of the resources failed and the provider has to adapt the system. For the use of client classification jobs where randomly determined to be from gold users with a probability of one third.

A resource oriented mechanism was used as benchmark to compare the performance of a purely technical mechanism with our mechanism which takes economic aspects into account. The mechanism used first checks which resource type has the highest capacity backlog. It then cancels the job requiring the most of this resource type. Subsequently it is checked whether capacity is sufficient. If this is not the case the mechanism is repeated until enough jobs are cancelled.

Table 1 shows the results of the evaluation of our basic, revenue maximizing, model. In our scenarios the runtimes for the calculation of the solutions were about a second. All numbers represent the average over all run simulations. It can be seen that our model delivers a revenue increase of about 36% compared the benchmark, while having a slightly lower ratio of jobs kept, i.e. cancelling a little more jobs on average. The lower ratio of jobs is caused by the fact that the benchmark mechanism tries to minimize the number of cancelled jobs by cancelling large jobs, while our model considers revenue regardless of job size.

	Revenue	Ratio jobs kept
Benchmark	1211	0,59
Basic Model RM	1652	0,54
Increase	+ 36,47%	-7,95%

Table 1. Results Basic Model RM

The results of the evaluation of the two extensions can be seen in Table 2:

	Revenue	Ratio jobs kept	Ratio gold jobs kept
Basic Model: RM	1652	0,54	0,48
1. Extension Rent	1858	0,62	0,60
2. Extension: CC	1241	0,54	1

Table 2. Results Basic Model and Extensions

The first extension, i.e. the ability to rent additional capacity, delivers a further increase in revenue. We assumed there is no limit in available capacity for rental in our simulations. However, the fees for rental where such that is was only profitable for higher paid jobs. Since there is more capacity available now, the first extension shows an increase in both the ratio of regular jobs kept as well as the ratio of gold jobs kept.

The second extension which focuses on client classification results in lower revenue than the basic model and the first extension. It only slightly outperforms the benchmark. It does however achieve its primary goal, which is to ensure that as many gold jobs are kept running as possible. In our scenarios there was always enough capacity left to accommodate all gold

users, therefore none of their jobs had to be cancelled. During the simulation a value of 10 was used for m. The overall ratio of jobs is the same as with the basic model.

The evaluation clearly shows the benefit of our model and its different applications. The basic model should be used for cloud provider who don't want give privileges for certain customers therefore focusing on (short-term) revenue maximization. Furthermore users of the basic model would not try rent additional capacity from other providers in overload situations. This could be the case for very complex services that cannot be easily migrated on external resources. The decision could also be influenced by data security and privacy issues as well as other business policies (e.g. not using competitors' services,).

The first extension of the basic model would be used by cloud providers without customer classification which can and are willing to make use of other providers' services if necessary.

The second extension is best for providers with internal users who need better service levels, key customers or framework contracts. Depending on the nature of the contracts and the revenue generated by internal users this model might also be able to outperform the basic model and first extension for revenue.

A combination of the first and second extension is also possible. In this case it would be possible to keep avoiding cancellation of gold jobs but further increase revenue compared to the second extension.

CONCLUSION

In this work a Service/Job Cancellation Model for dealing with partial resource failures and overload situations was motivated and proposed. Related work is described and analyzed in section 2. The proposed model and extensions are explained in section 3. Following the description of our evaluation scenario the results of the simulations are presented in the fourth section. The results show that revenue can be significantly increased by the use of our decision model. The basic model and first extension outperform the benchmark by over 30%. The first extension furthermore improves the ratio of accepted jobs, while the second extension focuses on minimizing cancellations for important customers. Depending on provider policies or framework contracts this can be a key advantage. It is further explained which model is best suitable for which situation.

Future work includes further evaluation of the model in different settings as well as evaluation of computational cost for large settings. It is expected that runtimes and computational cost can significantly increase for large of concurrently running jobs. Therefore, depending on the amount of resources and concurrently running jobs it might be necessary to use heuristic approaches instead of the calculation of optimal solutions. The introduction of further aspects of risk management in the model and its inclusion in SLAs will be another aspect for future work.

REFERENCES

1. Boughton, H., Martin, P., Powley, W., and Horman, R. (2006). "Workload class importance policy in autonomic database management systems," *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06),*Washington, DC, USA: IEEE Computer Society, pp. 13–22.
2. Buyya, R. (2002). *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, Ph.D. thesis, Monash University.
3. Carr, N. G. (2005). "The end of corporate computing," *MIT Sloan Management Review (46:3)*, pp. 32–42.
4. Chicco, G., Napoli, R., and Piglione, F. (2006). "Comparisons among clustering techniques for electricity customer classification," *IEEE Transactions on Power Systems (21:2)*, pp. 933– 940.
5. Djemame, K., Gourlay, I., Padgett, J., Birkenheuer, G., Hovestadt, M., Kao, O., and Voss, K. (2006). "Introducing risk management into the grid," *The 2nd IEEE International Conference on e-Science and Grid Computing (eScience2006)*, Amsterdam, Netherlands, p. 28.
6. Feitelson, D.G. (2009). "Workload Modeling for Computer Systems Evaluation".
7. Foster, I., Kesselman, C., Lee, C., Lindell, B., Nahrstedt, K., and Roy, A. (1999). "A distributed resource management architecture that supports advance reservations and co-allocation," *Proceedings of the 7th International Workshop on Quality of Service (IWQoS 1999)*, London, UK, pp. 62–80.

8. Gartner Inc. (2008). "Gartner Says Worldwide IT Spending On Pace to Surpass $3.4 Trillion in 2008," Press Release, http://www.gartner.com/it/page.jsp?id=742913, accessed on August, 21 2009.
9. Kenyon, C., and Cheliotis, G. (2004). "Grid resource commercialization: economic engineering and delivery scenarios," *Grid resource management: state of the art and future trends*, pp. 465–478.
10. Kounev, S., Nou, R., and Torres, J. (2007). "Building online performance models of grid middleware with fine-grained load-balancing: A globus toolkit case study," *The 4th European Engineering Performance Workshop (EPEW 2007)*, Berlin, Germany.
11. Newhouse, S., MacLaren, J., and Keahey, K. (2004). "Trading grid services within the uk e-science grid," *Grid resource management: state of the art and future trends*, pp. 479–490.
12. New York Times Blog (2007). TimesMachine use case, http://open.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/ (accessed on August, 21 2009)
13. Nou, R., Julià, F., and Torres, J. (2007). "Should the grid middleware look to selfmanaging capabilities?," *The 8th International Symposium on Autonomous Decentralized Systems (ISADS 2007)*, Sedona, Arizona, USA, pp. 113–122.
14. Poggi, N., Moreno, T., Berral, J. L., Gavaldà, R., and Torres, J. (2007). "Web customer modeling for automated session prioritization on high traffic sites," *Proceedings of the 11th International Conference on User Modeling*, Corfu, Greece.
15. Poggi, N., Moreno, T., Berral, J.L., Gavaldà, R., and Torres, J. (2009). "Self-adaptive utility-based web session management," *Comput. Netw. (53:10),* pp. 1712-1721.
16. Rappa, M. A. (2004). "The utility business model and the future of computing services," *IBM Systems Journal (43:1)*, pp. 32–42.
17. SmugMug Blog (2006). Amazon S3: Show me the money, http://blogs.smugmug.com/don/2006/11/10/amazon-s3-show-me-the-money/ (accessed on August, 21 2009)
18. Voss, K. (2006). "Risk-aware Migrations For Prepossessing SLAs," in *International conference on Networking and Services (ICNS '06)* Santa Clara, USA, July 2006, pp. 68–68.